Exhibit 20

https://learn.microsoft.com/en-us/legal/cognitive-services/openai/transparency-note?tabs=text

Transparency Note for Azure OpenAI Service

Article • 12/10/2024

What is a Transparency Note?

An AI system includes not only the technology, but also the people who use it, the people who are affected by it, and the environment in which it's deployed. Creating a system that is fit for its intended purpose requires an understanding of how the technology works, what its capabilities and limitations are, and how to achieve the best performance. Microsoft's Transparency Notes are intended to help you understand how our AI technology works, the choices system owners can make that influence system performance and behavior, and the importance of thinking about the whole system, including the technology, the people, and the environment. You can use Transparency Notes when developing or deploying your own system, or share them with the people who will use or be affected by your system.

Microsoft's Transparency Notes are part of a broader effort at Microsoft to put our AI Principles into practice. To find out more, see the Microsoft's AI principles .

The basics of the Azure OpenAI Models

Azure OpenAI provides customers with a fully managed AI service that lets developers and data scientists apply OpenAI's powerful models including models that can generate natural language, code, and images. Within the Azure OpenAI Service, the OpenAI models are integrated with Microsoft-developed content filtering and abuse detection models. Learn more about content filtering here and abuse detection here.

Introduction

Expand table

Model group	Text / code	Vision	Audio / Speech
GPT-3 & Codex	✓		

DALL-E 2 & 3		☑	
Whisper			☑
GPT-4 Turbo with Vision	☑	☑	
GPT-4o & GPT-4o-mini	☑	☑	☑
o1	☑	☑	

Select the tabs to see content for the relevant model type.

Text, code, and fine-tuned models

As part of the fully managed Azure OpenAI Service, the GPT-3 models analyze and generate natural language, Codex models analyze and generate code and plain text code commentary, and GPT-4 and o1 models can understand and generate natural language and code. These models use an autoregressive architecture, meaning they use data from prior observations to predict the most probable next word. This process is then repeated by appending the newly generated content to the original text to produce the complete generated response. Because the response is conditioned on the input text, these models can be applied to various tasks simply by changing the input text.

The GPT-3 series of models are pretrained on a wide body of publicly available free text data. This data is sourced from a combination of web crawling (specifically, a filtered version of Common Crawl , which includes a broad range of text from the internet and comprises 60 percent of the weighted pretraining dataset) and higher-quality datasets, including an expanded version of the WebText dataset, two internet-based books corpora and English-language Wikipedia. The GPT-4 base model was trained using publicly available data (such as internet data) and data that was licensed by OpenAI. The model was fine-tuned using reinforcement learning with human feedback (RLHF).

Learn more about the training and modeling techniques in OpenAI's GPT-3 , GPT-4 , and Codex research papers. The guidance below is also drawn from OpenAI's safety best practices .

Fine tuning refers to using *Supervised Fine Tuning* to adjust a base model's weights to provide better responses based on a provided training set. All use cases and considerations for large language models apply to fine-tuned models, but there are additional considerations as well.

ⓘ **Important**

Fine-tuning is only available for text and code models, not vision or speech models.

Key terms

Expand table

Term	Definition

Prompt	The text you send to the service in the API call. This text is then input into the model. For example, one might input the following prompt:
	`Convert the questions to a command:`
	`Q: Ask Constance if we need some bread`
	`A: send-msg 'find constance' Do we need some bread?`
	`Q: Send a message to Greg to figure out if things are ready for Wednesday.`
	`A:`

Completion or Generation	The text Azure OpenAI outputs in response. For example, the service may respond with the following answer to the above prompt: `send-msg 'find greg' figure out if things are ready for Wednesday.`
Token	Azure OpenAI processes text by breaking it down into tokens. Tokens can be words or just chunks of characters. For example, the word `hamburger` gets broken up into the tokens `ham`, `bur` and `ger`, while a short and common word like `pear` is a single token. Many tokens start with a whitespace, for example `hello` and `bye`.
Fine tuning	Supervised fine-tuning for large language models refers to the process of taking a pre-trained language model, often trained on a massive dataset, and further training it on a more specific task with labeled data. This involves adjusting the weights of the model using this smaller, specific dataset so that the model becomes more specialized in the tasks it can perform, enhancing its performance and accuracy.
Model Weights	Model weights are parameters within the model that are learned from the data during the training process. They determine the output of the model for a

Term	Definition
	given input. These weights are adjusted in response to the error the model made in its predictions, with the aim of minimizing this error.
Ungrounded content	Content that is generated by the model that is non-factual or inaccurate from what was present in the source materials.

Capabilities

Text, code, and fine-tuned models

The o1, GPT-4, GPT-3, and Codex models, and Azure OpenAI evaluation use natural language instructions and examples in the prompt to identify the task. The model then completes the task by predicting the most probable next text. This technique is known as "in-context" learning. These models are not retrained during this step but instead give predictions based on the context you include in the prompt.

There are three main approaches for in-context learning. These approaches vary based on the amount of task-specific data that is given to the model:

Few-shot : In this case, a user includes several examples in the prompt that demonstrate the expected answer format and content. The following example shows a few-shot prompt providing multiple examples:

```
Convert the questions to a command:
Q: Ask Constance if we need some bread
A: send-msg `find constance` Do we need some bread?
Q: Send a message to Greg to figure out if things areready for Wednesday.
A: send-msg `find greg` Is everything ready forWednesday?
Q: Ask Ilya if we're still having our meeting thisevening
A: send-msg `find ilya` Are we still having a meetingthis evening?
Q: Contact the ski store and figure out if I can getmy skis fixed before I
leave on Thursday
A: send-msg `find ski store` Would it be possible toget my skis fixed be-
fore I leave on Thursday?
Q: Thank Nicolas for lunch
A: send-msg `find nicolas` Thank you for lunch!
Q: Tell Constance that I won't be home before 19:30tonight — unmovable
meeting.
A: send-msg `find constance` I won't be home before19:30 tonight. I have a
```

```
meeting I can't move.
Q: Tell John that I need to book an appointment at10:30
A:
```

The number of examples typically ranges from 0 to 100 depending on how many can fit in the maximum input length for a single prompt. Few-shot learning enables a major reduction in the amount of task-specific data required for accurate predictions.

One-shot : This case is the same as the few-shot approach except only one example is provided. The following example shows a one-shot prompt:

```
Convert the questions to a command:
Q: Ask Constance if we need some bread
A: send-msg `find constance` Do we need some bread?
Q: Send a message to Greg to figure out if things are ready for Wednesday.
A:
```

Zero-shot: In this case, no examples are provided to the model and only the task request is provided. The following example shows a zero-shot prompt:

```
Convert the question to a command:
Q: Ask Constance if we need some bread
A:
```

Chain-of-thought : Azure OpenAI's o1-preview and o1-mini models have new advanced reasoning capabilities using chain-of-thought (CoT) techniques. CoT techniques generate intermediate reasoning steps before providing a response, enabling o1 to address more complex challenges through step-by-step problem solving. o1 demonstrates improvements in benchmarks for reasoning-heavy domains such as research, strategy, science, coding and math, among others. o1 has safety improvements from advanced reasoning capabilities, with the ability to reason through and apply safety rules more effectively. This results in better performance alongside safety benchmarks such as generating illicit advice, choosing stereotyped responses, and succumbing to known jailbreaks.

For greater detail on the o1 family of models' capabilities, see the OpenAI o1 System Card .

Azure OpenAI Evaluation

The evaluation of large language models is a critical step in measuring their performance across various tasks and dimensions. This task is especially important for fine-tuned models, where assessing the performance gains (or losses) from training is crucial. Without thorough evaluations, it can become challenging to understand how different versions of the model may impact your specific application .

Azure OpenAI Evaluation is a UI-based experience to evaluate data, including generated datasets from an Azure OpenAI deployment, or other manually curated files.

Azure OpenAI Evaluation has an optional step of generating responses. If the user opts into this step, we provide a prompt (System/User Message) to instruct the model how to generate responses.

Azure OpenAI Evaluation includes 9 categories of tests to score results. Some require ground truth data (like factuality), while others do not (schema validation). Graders are a mixture of CPU-based and model-based. Here is the list of testing criteria: Factuality, Sentiment, Valid JSON or XML, Criteria Match, Custom Prompt, Semantic Similarity, Contains string, Matches Schema and Text quality.

Use cases

Text, code, and fine-tuned models

Intended uses

The o1, GPT-4, GPT-3, and Codex models, and Azure OpenAI evaluation can be used in multiple scenarios. The following list isn't comprehensive, but it illustrates the diversity of tasks that can be supported for models with appropriate mitigations:

- **Chat and conversation interaction** : Users can interact with a conversational agent that responds with responses drawn from trusted documents such as internal company documentation or tech support documentation. Conversations must be limited to answering scoped questions.

- **Chat and conversation creation** : Users can create a conversational agent that responds with responses drawn from trusted documents such as internal company documentation or tech support documentation. Conversations must be limited to answering scoped questions.
- **Code generation or transformation scenarios** : For example, converting one programming language to another, generating docstrings for functions, converting natural language to SQL.
- **Journalistic content** : For use to create new journalistic content or to rewrite journalistic content submitted by the user as a writing aid for predefined topics. Users cannot use the application as a general content creation tool for all topics.
- **Question-answering** : Users can ask questions and receive answers from trusted source documents such as internal company documentation. The application does not generate answers ungrounded in trusted source documentation.
- **Reason over structured and unstructured data** : Users can analyze inputs using classification, sentiment analysis of text, or entity extraction. Examples include analyzing product feedback sentiment, analyzing support calls and transcripts, and refining text-based search with embeddings.
- **Search** : Users can search trusted source documents such as internal company documentation. The application does not generate results ungrounded in trusted source documentation.
- **Summarization** : Users can submit content to be summarized for predefined topics built into the application and cannot use the application as an open-ended summarizer. Examples include summarization of internal company documentation, call center transcripts, technical reports, and product reviews.
- **Writing assistance on specific topics** : Users can create new content or rewrite content submitted by the user as a writing aid for business content or pre-defined topics. Users can only rewrite or create content for specific business purposes or predefined topics and cannot use the application as a general content creation tool for all topics. Examples of business content include proposals and reports. For journalistic use, see above **Journalistic content** use case.
- **Data generation for fine-tuning:** Users can use a model in Azure OpenAI to generate data which is used solely to fine-tune (i) another Azure OpenAI model, using the fine-tuning capabilities of Azure OpenAI, and/or (ii) another Azure AI custom model, using the fine-tuning capabilities of the Azure AI service. Generating data and fine-tuning models is limited to internal users only; the fine-tuned model may only be used for inferencing in the applicable Azure AI service

and, for Azure OpenAI service, only for customer's permitted use case(s) under this form.

Fine-tuned use cases

The following are additional use cases we recommend for fine-tuned text and code models. Fine tuning is most appropriate for:

- **Steering the style, format, tone or qualitative aspects of responses** via examples of the desired responses.
- **Ensuring the model reliably produces a desired output** such as providing responses in a specific format or ensuring responses are grounded by information in the prompt.
- **Use cases with many edge cases** that cannot be covered within examples in the prompt, such as complex natural language to code examples.
- **Improving performance at specific skills or tasks** such as classification, summarization, or formatting – that can be hard to describe within a prompt.
- **Reducing costs or latency** by utilizing shorter prompts, or swapping a fine-tuned version of a smaller/faster model for a more general-purpose model (e.g. fine tuned GPT-3.5-Turbo for GPT-4).

As with base models, the use case prohibitions outlined in the Azure OpenAI Code of conduct apply to fine-tuned models as well.

Fine tuning alone is not recommended for scenarios where you want to extend your model to include out-of-domain information, where explainability or grounding are important, or where the underlying data are updated frequently.

o1 use cases

The advanced reasoning capabilities of the o1-preview and o1-mini models may be best suited for reasoning-heavy uses in science, coding, math, and similar fields. Specific use cases could include:

- **Complex code generation, analysis and optimization**: Algorithm generation and advanced coding tasks to help developers execute multi-step workflows, better understanding the steps taken in code development.
- **Advanced problem solving**: Comprehensive brainstorming sessions, strategy development and breaking down multifaceted issues.

- **Complex document comparison**: Analyzing contracts, case files, or legal documents to discern subtle differences in document contents.
- **Instruction following and workflow management**: Handling workflows that require shorter context.

For greater detail on o1's intended uses, visit the OpenAI o1 System Card .

Azure OpenAI evaluation use cases

Azure OpenAI evaluation is a text-only feature and can't be used with models that support non-text inputs. Evals can be used in multiple scenarios including but not limited to:

- **Text matching/comparison evaluation**: This is helpful for scenarios where the user wants to check if the output matches an expected string. Users can also compare two sets of values and score the relationships. Examples include, but are not limited to, multiple-choice questions where answers are compared to an answer key, and string validation.
- **Text quality**: Text quality assesses response quality with methods such as Bleu, Rouge or cosine algorithms and is widely used in various natural language processing tasks such as machine translation, text summarization, and text generation, among others.
- **Classification-based evaluation**: Classification-based evaluation assesses the performance of a model by assigning responses to predefined categories or labels or by comparing the model's output to a reference set of correct answers. Automated grading, sentiment analysis, and product categorization are among some of the common use cases.
- **Conversational quality evaluation**: Conversational quality evaluation involves comparing responses against predefined criteria using a detailed chain-of-thought (CoT) prompt. Common use cases include customer support, chatbot development, and educational assessments, among others.
- **Criteria-based evaluation**: One common scenario for criteria-based evaluation is factuality. Assessing factual accuracy involves comparing a submitted answer to an expert answer, focusing solely on factual content. This can be useful in educational tools to improve the accuracy of answers provided by LLMs or in research assistance tools to assess the factual accuracy of responses generated by LLMs in academic settings.
- **String validity evaluation**: one common scenario would be to check if model's response follows a specific schema or is valid JSON or XML content.

Considerations when choosing a use case

We encourage customers to use the Azure OpenAI GPT-4, GPT-3, and Codex models in their innovative solutions or applications as approved in their Limited Access registration form. However, here are some considerations when choosing a use case:

- **Not suitable for open-ended, unconstrained content generation.** Scenarios where users can generate content on any topic are more likely to produce offensive or harmful text. The same is true of longer generations.
- **Not suitable for scenarios where up-to-date, factually accurate information is crucial** unless you have human reviewers or are using the models to search your own documents and have verified suitability for your scenario. The service does not have information about events that occur after its training date, likely has missing knowledge about some topics, and may not always produce factually accurate information.
- **Avoid scenarios where use or misuse of the system could result in significant physical or psychological injury to an individual.** For example, scenarios that diagnose patients or prescribe medications have the potential to cause significant harm. Incorporating meaningful human review and oversight into the scenario can help reduce the risk of harmful outcomes.
- **Avoid scenarios where use or misuse of the system could have a consequential impact on life opportunities or legal status.** Examples include scenarios where the AI system could affect an individual's legal status, legal rights, or their access to credit, education, employment, healthcare, housing, insurance, social welfare benefits, services, opportunities, or the terms on which they're provided. Incorporating meaningful human review and oversight into the scenario can help reduce the risk of harmful outcomes.
- **Avoid high stakes scenarios that could lead to harm.** The models hosted by Azure OpenAI service reflect certain societal views, biases, and other undesirable content present in the training data or the examples provided in the prompt. As a result, we caution against using the models in high-stakes scenarios where unfair, unreliable, or offensive behavior might be extremely costly or lead to harm. Incorporating meaningful human review and oversight into the scenario can help reduce the risk of harmful outcomes.
- **Carefully consider use cases in high stakes domains or industry:** Examples include but are not limited to healthcare, medicine, finance, or legal.
- **Carefully consider well-scoped chatbot scenarios.** Limiting the use of the service in chatbots to a narrow domain reduces the risk of generating unintended or

undesirable responses.

- **Carefully consider all generative use cases.** Content generation scenarios may be more likely to produce unintended outputs and these scenarios require careful consideration and mitigations.
- **Legal and regulatory considerations:** Organizations need to evaluate potential specific legal and regulatory obligations when using any AI services and solutions, which may not be appropriate for use in every industry or scenario. Additionally, AI services or solutions are not designed for and may not be used in ways prohibited in applicable terms of service and relevant codes of conduct.

Limitations

When it comes to large-scale natural language models, vision models, and speech models, there are fairness and responsible AI issues to consider. People use language and images to describe the world and to express their beliefs, assumptions, attitudes, and values. As a result, publicly available text and image data typically used to train large-scale natural language processing and image generation models contains societal biases relating to race, gender, religion, age, and other groups of people, as well as other undesirable content. Similarly, speech models can exhibit different levels of accuracy across different demographic groups and languages. These societal biases are reflected in the distributions of words, phrases, and syntactic structures.

Technical limitations, operational factors, and ranges

⊗ **Caution**

Be advised that this section contains illustrative examples which include terms and language that some individuals might find offensive.

Large-scale natural language, image, and speech models trained with such data can potentially behave in ways that are unfair, unreliable, or offensive, in turn causing harms. Some of the ways are listed here. We emphasize that these types of harms are not mutually exclusive. A single model can exhibit more than one type of harm, potentially relating to multiple different groups of people. For example:

- **Allocation:** These models can be used in ways that lead to unfair allocation of resources or opportunities. For example, automated résumé screening systems can withhold employment opportunities from one gender if they are trained on résumé data that reflects the existing gender imbalance in a particular industry. Or the DALL·E models could be used to create imagery in the style of a known artist, which could affect the value of the artist's work or the artist's life opportunities. GPT-4 vision models could be used to identify individual behaviors and patterns that might have negative impacts on life opportunities.
- **Quality of service:** The Azure OpenAI models are trained primarily on English text and images with English text descriptions. Languages other than English will experience worse performance. English language varieties with less representation in the training data might experience worse performance than standard American English. The publicly available images used to train the DALL·E models might reinforce public bias and other undesirable content. The DALL·E models are also unable to consistently generate comprehensible text at this time. Speech models might introduce other limitations, for example, translations using the Whisper model in Azure OpenAI are limited to English output only. Broadly speaking, with Speech-to-Text models, be sure to properly specify a language (or locale) for each audio input to improve accuracy in transcription. Additionally, acoustic quality of the audio input, non-speech noise, overlapped speech, vocabulary, accents, and insertion errors might also affect the quality of your transcription or translation.
- **Stereotyping:** These models can reinforce stereotypes. For example, when translating "He is a nurse" and "She is a doctor" into a genderless language such as Turkish and then back into English, many machine translation systems yield the stereotypical (and incorrect) results of "She is a nurse" and "He is a doctor." With DALL·E, when generating an image based on the prompt "Fatherless children," the model could generate images of Black children only, reinforcing harmful stereotypes that might exist in publicly available images. The GPT-4 vision models might also reinforce stereotypes based on the contents of the input image, by relying on components of the image and making assumptions that might not always be true.
- **Demeaning:** The natural language and vision models in the Azure OpenAI service can demean people. For example, an open-ended content generation system with inappropriate or insufficient mitigations might produce content that is offensive or demeaning to a particular group of people.
- **Overrepresentation and underrepresentation:** The natural language and vision models in the Azure OpenAI service can over- or under-represent groups of people, or even erase their representation entirely. For example, if text prompts that contain the word "gay" are detected as potentially harmful or offensive, this identification

could lead to the underrepresentation or even erasure of legitimate image generations by or about the LGBTQIA+ community.

- **Inappropriate or offensive content:** The natural language and vision models in the Azure OpenAI service can produce other types of inappropriate or offensive content. Examples include the ability to generate text that is inappropriate in the context of the text or image prompt;, the ability to create images that potentially contain harmful artifacts such as hate symbols; images that illicit harmful connotations; images that relate to contested, controversial, or ideologically polarizing topics; images that are manipulative; images that contain sexually charged content that is not caught by sexual-related content filters; and images that relate to sensitive or emotionally charged topics. For example, a well-intentioned text prompt aimed to create an image of the New York skyline with clouds and airplanes flying over it might unintentionally generate images that illicit sentiments related to the events surrounding 9/11.
- **Disinformation and misinformation about sensitive topics:** Because DALL·E 2 and DALL·E 3 are powerful image generation models, they can be used to produce disinformation and misinformation that can be harmful. For example, a user could prompt the model to generate an image of a political leader engaging in activity of a violent or sexual (or simply inaccurate) nature that might lead to consequential harms, including but not limited to public protests, political change, or fake news. The GPT-4 visions models could also be used in a similar vein. The model might reinforce disinformation or misinformation about sensitive topics if the prompt contains such information without mitigation.
- **Information reliability:** Language and vision model responses can generate nonsensical content or fabricate content that might sound reasonable but is inaccurate with respect to external validation sources. Even when drawing responses from trusted source information, responses might misrepresent that content. Transcriptions or translations might result in inaccurate text.
- **False information:** Azure OpenAI does not fact-check or verify content that is provided by customers or users. Depending on how you have developed your application, it might produce false information unless you have built in mitigations (see **Best practices for improving system performance**).

Risks and limitations of fine-tuning

When customers fine-tune Azure OpenAI models, it can improve model performance and accuracy on specific tasks and domains, but it can also introduce new risks and limitations that customers should be aware of. These risks and limitations apply to all Azure OpenAI models that support fine-tuning. Some of these risks and limitations are:

- **Data quality and representation**: The quality and representativeness of the data used for fine-tuning can affect the model's behavior and outputs. If the data is noisy, incomplete, outdated, or if it contains harmful content like stereotypes, the model can inherit these issues and produce inaccurate or harmful results. For example, if the data contains gender stereotypes, the model can amplify them and generate sexist language. Customers should carefully select and pre-process their data to ensure that it is relevant, diverse, and balanced for the intended task and domain.
- **Model robustness and generalization**: The model's ability to handle diverse and complex inputs and scenarios can decrease after fine-tuning, especially if the data is too narrow or specific. The model can overfit to the data and lose some of its general knowledge and capabilities. For example, if the data is only about sports, the model can struggle to answer questions or generate text about other topics. Customers should evaluate the model's performance and robustness on a variety of inputs and scenarios and avoid using the model for tasks or domains that are outside its scope.
- **Regurgitation**: While your training data is not available to Microsoft or any third-party customers, poorly fine-tuned models may regurgitate, or directly repeat, training data. Customers are responsible for removing any PII or otherwise protected information from their training data and should assess their fine-tuned models for over-fitting or otherwise low-quality responses. To avoid regurgitation, customers are encouraged to provide large and diverse datasets.
- **Regurgitation**: While your training data is not available to Microsoft or any third-party customers, poorly fine-tuned models may regurgitate, or directly repeat, training data. Customers are responsible for removing any PII or otherwise protected information from their training data and should assess their fine-tuned models for over-fitting or otherwise low-quality responses. To avoid regurgitation, customers are encouraged to provide large and diverse datasets.
- **Model transparency and explainability**: The model's logic and reasoning can become more opaque and difficult to understand after fine-tuning, especially if the data is complex or abstract. A fine-tuned model can produce outputs that are unexpected, inconsistent, or contradictory, and customers may not be able to explain how or why the model arrived at those outputs. For example, if the data is about legal or medical terms, the model can generate outputs that are inaccurate or misleading, and customers may not be able to verify or justify them. Customers should monitor and audit the model's outputs and behavior and provide clear and accurate information and guidance to the end-users of the model.

To help mitigate the risks associated with advanced fine-tuned models, we have implemented additional evaluation steps to help detect and prevent harmful content in the training and outputs of fine-tuned models. The fine-tuned model evaluation filters are set to predefined thresholds and cannot be modified by customers; they aren't tied to any custom content filtering configuration you may have created.

o1 limitations

- o1 is best suited for use cases that involve heavy reasoning and may not perform well on some natural language tasks such as personal or creative writing when compared to earlier AOAI models.
- o1's new CoT reasoning capabilities may increase certain types of risks, requiring refined methods and approaches towards risk management protocols and evaluating and monitoring system behavior. For example, o1's CoT reasoning capabilities have demonstrated improvements in persuasiveness, and simple in-context scheming.
- Users may experience that the o1 family of models takes more time to reason through responses and should account for the additional time and latency in developing applications.

For greater detail on o1's limitations, see the OpenAI o1 System Card .

4o limitations

- The `gpt-4o-realtime-preview` audio translation capabilities may output non-English languages in a non-native accent. This may limit the effectiveness of language performance in audio outputs. Language supportability is in line with existing gpt-4o model versions.
- Users may experience that `gpt-4o-realtime-preview` is less robust in noisy environments and should account for noise sensitivity when developing applications.

For more best practices, see the OpenAI 4o System Card .

Azure OpenAI evaluation limitations

- **Data Quality**: When you're using Azure OpenAI Evaluation, be aware that poor quality data can lead to misleading or unreliable evaluation results.
- **Configuration quality:** If a customer improperly defines the prompt or evaluators or provides invalid evaluation data, the results of the Azure OpenAI Evaluation service will be incorrect and invalid. Refer to the Azure OpenAI documentation for details on how to set up an evaluation run.
- **Limited scope**: Azure OpenAI evaluation only supports text-based natural language models. It does not support any risk and safety metrics to evaluate generated responses for risk and safety severity scores (e.g., hateful and unfair content, sexual content, violent content, and self-harm related content).

System performance

In many AI systems, performance is often defined in relation to accuracy—that is, how often the AI system offers a correct prediction or output. With large-scale natural language models and vision models, two different users might look at the same output and have different opinions of how useful or relevant it is, which means that performance for these systems must be defined more flexibly. Here, we broadly consider performance to mean that the application performs as you and your users expect, including not generating harmful outputs.

Azure OpenAI service can support a wide range of applications like search, classification, code generation, image generation, and image understanding, each with different performance metrics and mitigation strategies. There are several steps you can take to mitigate some of the concerns listed under "Limitations" and to improve performance. Other important mitigation techniques are outlined in the section Evaluating and integrating Azure OpenAI for your use.

Best practices for improving system performance

- **Show and tell when designing prompts.** With natural language models and speech models, make it clear to the model what kind of outputs you expect through instructions, examples, or a combination of the two. If you want the model to rank a list of items in alphabetical order or to classify a paragraph by sentiment, show the model that is what you want.
    - **Prompts for the Whisper model in Azure OpenAI service** can help improve model outputs. The following best practices will help you create prompts that best fit your scenario and needs.
        - Consider including a prompt to instruct the model to correct specific words or acronyms that the model often misrecognizes in the audio.
        - To preserve the context of a file that was split into segments, you might prompt the model with the transcript of the preceding segment. This prompt will make the transcript more accurate, because the model will use the relevant information from the previous audio. The model will only consider the final 224 tokens of the prompt and ignore anything earlier.
        - The model might skip punctuation in the transcript. Consider using a simple prompt that instructs the model to include punctuation.
        - The model might also leave out common filler words, for example, hmmm, umm, etc. in the audio. If you want to keep the filler words in your transcript,

you might include a prompt that contains them.

  - Some languages can be written in different ways, such as simplified or traditional Chinese. The model might not always use the writing style that a user wants for their transcript by default. Consider using a prompt to describe your preferred writing style.

- **Keep your application on topic.** Carefully structure prompts and image inputs to reduce the chance of producing undesired content, even if a user tries to use it for this purpose. For instance, you might indicate in your prompt that a chatbot only engages in conversations about mathematics and otherwise responds "I'm sorry. I'm afraid I can't answer that." Adding adjectives like "polite" and examples in your desired tone to your prompt can also help steer outputs. With DALL·E models, you might indicate in your prompt or image input that your application generates only conceptual images. It might otherwise generate a pop-up notification that explains that the application is not for photorealistic use or to portray reality. Consider nudging users toward acceptable queries and image inputs, either by listing such examples up front or by offering them as suggestions upon receiving an off-topic request. Consider training a classifier to determine whether an input (prompt or image) is on topic or off topic.
- **Provide quality data.** With text and code models, if you are trying to build a classifier or get the model to follow a pattern, make sure that there are enough examples. Be sure to proofread your examples—the model is usually capable of processing basic spelling mistakes and giving you a response, but it also might assume errors are intentional which could affect the response. Providing quality data also includes giving your model reliable data to draw responses from in chat and question answering systems.
- **Provide trusted data.** Retrieving or uploading untrusted data into your systems could compromise the security of your systems or applications. To mitigate these risks in your applicable applications (including applications using the Assistants API), we recommend logging and monitoring LLM interactions (inputs/outputs) to detect and analyze potential prompt injections, clearly delineating user input to minimize risk of prompt injection, restricting the LLM's access to sensitive resources, limiting its capabilities to the minimum required, and isolating it from critical systems and resources. Learn about additional mitigation approaches in Security guidance for Large Language Models | Microsoft Learn.
- **Configure parameters to improve accuracy or groundedness of responses.** Augmenting prompts with data retrieved from trusted sources – such as by using the Azure OpenAI "on your data" feature – can reduce, but not completely eliminate, the likelihood of generating inaccurate responses or false information. Steps you can take

to further improve the accuracy of responses include carefully selecting the trusted and relevant data source and configuring custom parameters such as "strictness", "limit responses to data content" and "number of retrieved documents to be considered" as appropriate to your use cases or scenarios. Learn more about configuring these settings for Azure OpenAI on Your Data.

- **Measure model quality.** As part of general model quality, consider measuring and improving fairness-related metrics and other metrics related to responsible AI in addition to traditional accuracy measures for your scenario. Consider resources like this checklist when you measure the fairness of the system. These measurements come with limitations, which you should acknowledge and communicate to stakeholders along with evaluation results.
- **Limit the length, structure, and rate of inputs and outputs.** Restricting the length or structure of inputs and outputs can increase the likelihood that the application will stay on task and mitigate, at least in part, any potentially unfair, unreliable, or offensive behaviour. Other options to reduce the risk of misuse include (i) restricting the source of inputs (for example, limiting inputs to a particular domain or to authenticated users rather than being open to anyone on the internet) and (ii) implementing usage rate limits.
- **Encourage human review of outputs prior to publication or dissemination.** With generative AI, there is potential for generating content that might be offensive or not related to the task at hand, even with mitigations in place. To ensure that the generated output meets the task of the user, consider building ways to remind users to review their outputs for quality prior to sharing widely. This practice can reduce many different harms, including offensive material, disinformation, and more.
- **Implement additional scenario-specific mitigations.** Refer to the mitigations outlined in Evaluating and integrating Azure OpenAI for your use including content moderation strategies. These recommendations do not represent every mitigation required for your application. Newer models such as GPT-4o and o1 may provide responses in sensitive scenarios and are more likely to attempt to reduce potentially harmful outputs in their responses rather than refuse to respond altogether. It is important to understand this behavior when evaluating and integrating content moderation for your use case; adjustments to filtering severity may be needed depending on your use case.
- o1 models have safeguards to prevent output of raw CoT. Attempting to extract the raw CoT, for example, by circumventing these safeguards, violates the Acceptable Use Policy for Online Services and may result in access to the service being limited. For greater detail on best practices, visit the OpenAI o1 System card .

Best practices and recommendations for fine tuning

To mitigate the risks and limitations of fine-tuning models on Azure OpenAI, we recommend customers to follow some best practices and guidelines, such as:

- **Data selection and preprocessing:** Customers should carefully select and pre-process their data to ensure that it is relevant, diverse, and balanced for the intended task and domain. Customers should also remove or anonymize any sensitive or personal information from the data, such as names, addresses, or email addresses, to protect the privacy and security of the data subjects. Customers should also check and correct any errors or inconsistencies in the data, such as spelling, grammar, or formatting, to improve the data quality and readability.
- **Include a system message in your training data** for chat-completion formatted models, to steer your responses, and use that same system message when using your fine-tuned model for inferencing. Leaving the system message blank tends to produce low-accuracy fine-tuned models, and forgetting to include the same system message when inferencing may result in the fine-tuned model reverting to the behavior of the base model.
- **Model evaluation and testing: Customers should evaluate and test the fine-tuned model's performance** and robustness on a variety of inputs and scenarios and compare it with the original model and other baselines. Customers should also use appropriate metrics and criteria to measure the model's accuracy, reliability, and fairness, and to identify any potential errors or biases in the model's outputs and behavior.
- **Model documentation and communication:** Customers should document and communicate the model's purpose, scope, limitations, and assumptions, and provide clear and accurate information and guidance to the end-users of the model.

Best practices and recommendations for Azure OpenAI evaluation

- **Robust ground truth data:** In general in large-scale natural language models, customers should carefully select and pre-process their data to ensure that it is relevant, diverse, and balanced for the intended task and domain. Customers should also remove or anonymize any sensitive or personal information from the data, such as names, addresses, or email addresses, to protect the privacy and security of the data subjects. Customers should also check and correct any errors or inconsistencies in the data, such as spelling, grammar, or formatting, to improve the data quality and readability.

Specifically for Azure OpenAI evaluation, the accuracy of the ground truth data provided by the user is crucial because inaccurate ground truth data leads to meaningless and inaccurate evaluation results. Ensuring the quality and reliability of this data is essential for obtaining valid assessments of the model's performance. Inaccurate ground truth data can skew the evaluation metrics, resulting in misleading conclusions about the model's capabilities. Therefore, users must carefully curate and verify their ground truth data to ensure that the evaluation process accurately reflects the model's true performance. This is particularly important when making decisions about deploying the model in real-world applications

- **Prompt definition for evaluation:** The prompt you use in your evaluation should match the prompt you plan to use in production. These prompts provide the instructions for the model to follow. Similar to the OpenAI playground, you can create multiple inputs to include few-shot examples in your prompt. Refer to Prompt engineering techniques for more details on some advanced techniques in prompt design and prompt engineering.
- **Diverse metrics**: Use a combination of metrics to capture different aspects of performance such as accuracy, fluency and relevance.
- **Human-in-the-loop**: Integrate human feedback alongside automated evaluation to ensure that subjective nuances are accurately captured.
- **Transparency**: Clearly communicate the evaluation criteria to users, enabling them to understand how decisions are made.
- **Continual evaluation and testing**: Continually evaluate the model's performance to identify and address any regressions or negative user experience.

Evaluating and integrating Azure OpenAI natural language and vision models for your use

Text, code, and fine-tuned models

The steps in conducting an Azure OpenAI evaluation are:

1. **Provide data for evaluation**: Either an uploaded flat file in JSONL format, or generated data based on a series of prompts.
2. **Specify test cases to evaluate the data**: Select one or more test cases to score the provided data with passing / failing grades.
3. **Review and filter results**: Each test includes a definition of passing and failing scores. After an evaluation runs, users can review their row-by-row results to see

individual test results, or filter on passed / failed.

For additional information on how to evaluate and integrate these models responsibly, please see the RAI Overview document.

Learn more about responsible AI

- Microsoft AI principles
- Microsoft responsible AI resources
- Microsoft Azure Learning courses on responsible AI

Learn more about Azure OpenAI

- Limited access to Azure OpenAI Service - Azure AI services | Microsoft Learn
- Code of Conduct for the Azure OpenAI Service | Microsoft Learn
- Data, privacy, and security for Azure OpenAI Service - Azure AI services | Microsoft Learn